Hacking CubeSats,
Cybersecurity in Orbit.

by Patrick H. Stakem

Number 6 in the Cubesat Series

Table of Contents

Introduction

This book builds upon computer security, to define security for small spacecraft. We will focus on our newest orbital members, Cubesats. There is a brief introduction on Cubesats and their command control system and control center. Then we will look at cyber attack story's from the commercial and government experience. We will then suggest some updates and new design approached for upcoming Cubesat mission.

We will not specifically discuss cyber attacks or protections in the military field. Leave that to the Space Force.

In 1998, hackers took over the NASA satellite ROSAT, a joint mission with the Germans. They had hacked into the Control Center at the Goddard Space Flight Center, and took over. They oriented the solar panels onto the Sun, and fried the batterys. In 1999, Hackers took over the U.K.'s military communications constellation, SkyNet. Supposedly, the unknown group issued blackmail threats against the Ministry of Defense. This has not been verified. There have been other supposedly satellite-hacking attacks by groups of state-sponsored terrorism.

Author

The author has a BSEE in Electrical Engineering from Carnegie-Mellon University, and Masters Degrees in Applied Physics and Computer Science from the Johns Hopkins University. During a career as a NASA support contractor from 1971 to 2013, he worked at all of the NASA Centers. He served as a mentor for the NASA/GSFC Summer Robotics Engineering Boot Camp at GSFC for 2 years.

Mr. Stakem has been affiliated with the Whiting School of Engineering of the Johns Hopkins University since 2007. He supported two summer Cubesat sessions for selected students under the Brazilian Scientific Mobility Program.

What is a Cubesat?

A Cubesat is a small, affordable satellite that can be developed and launched by college, high schools, and even individuals. The specifications were developed by Academia in 1999. The basic structure is a 10 centimeter cube, (volume of 1 liter) weighing less than 1.33 kilograms. This allows multiples of these standardized packages to be launched as secondary payloads on other missions. A Cubesat dispenser has been developed, the Poly-PicoSat Orbital Deployer, P-POD, that holds multiple Cubesats and dispenses them on orbit. They can also be launched from the Space Station, via a custom airlock. ESA, the United States, Russia, and others provide launch services. The Cubesat origin lies with Prof. Twiggs of Stanford University and was proposed as a vehicle to support hands-on university-level space education and opportunities for low-cost space access. This was at a presentation at the University Space Systems Symposium in Hawaii in November of 1999.

Cubesats began as teaching tools, and remain in that role, although their vast numbers in orbit showed they they have become mainstream. They have been launched by Corporations such as Space-X, wh has a constellation of 242 in orbit. They hope to launch thousands over the next few years. One goal is to provide the Internet to everywhere on the planet. This defines the challenge, then, to protect those assets from threats and hackers. As we become more and more dependent on the Internet for communications, entertainment, and as a source of information, we need increasing security.

In what has been called the Revolution of Smallsats, Cubesats lead the way. They represent paradigm shifts in developing space missions, opening the field from National efforts and large Aerospace contractors, to individuals and schools.

Even if your personal Cubesat project never gets launched or even built, it will bring you valuable experience to participate. This book will introduce and explain the NASA Systems Engineering Process, which leads from a set of goals to a successful space flight. This model has been in use for decades, and has proven

itself to have usefulness. It has been refined as problems were uncovered, and still remains a viable approach to space missions, as well as regular engineering applications. Security is a part of the process.

Cubesats can be custom made, but a major industry has evolved to supply components, including space computers. It allows for an off-the-shelf implementation, in addition to the custom build.

Build costs can be lower than $10,000, with launch costs ranging around $100,000, a most cost-effective price for achieving orbit. The low orbits of the Cubesats insure eventual reentry into the atmosphere, so they do not contribute to the orbital debris problem.

Central to the Cubesat concept is the standardization of the interface between the launch vehicle and the spacecraft, which allows developers to pool together for launch and so reduce costs and increase opportunities. As a university-led initiative, Cubesat developers have advocated many cost-saving mechanisms.

A simple Cubesat flight controller can be developed from a standard embedded computing platform such as the Arduino or Raspberry Pi. The lack of radiation hardness can be balanced by the short on-orbit lifetime. The main drivers for a Cubesat flight computer are small size, small power consumption, wide functionality, and flexibility. In addition, a wide operating temperature range is desirable. The architecture should support a real time operating system, but, in the simplest case, a simple loop program with interrupt support can work.

Earth imaging is a common objective for a Cubesat mission, typically achieved using a CMOS camera without any complex lens systems. As it is a critical impediment to the development of a highly capable platform for mission operations, the testing and evaluation of novel approaches for increasing downlink data rate and reliability is also a common objective. While less common than Earth imaging, real science objectives are becoming increasingly popular as recognition (primarily by NASA) of Cubesat capabilities increase and collaborations between

engineering and science groups emerge.

The Cubesat Design Specification

The Cubesat Design Specification, developed by California Polytechnic State University, defines the physical and interface specifications for Cubesats, and gives testing requirements for vibration, thermal-vacuum tests, and shock, as well as safety. Since a Cubesat flys with other Cubesats in a deployment device, and with a primary payload, safety is a concern. Cubesats are expected to have an on-orbit lifetime of less than 30 years.

There are emerging standards for larger Cubesats, such as 6U (12 kg, 12 x 24 x 36 cm), 12U (24 kg, 23 x 24 x 36 cm) and 27U (54 kg, 34 x 35 x 36 cm). These allow the canister to constrain Cubesat deployables such as antennae and solar array. In the original Cubesat specification, this task had to be handled by the Cubesat itself. Even as they get bigger, the standard architecture and modularity of the Cubesat remains a game-changing advantage.

There are currently no requirements for security of transmissions, or encryption of the uplink.

Although the Raspberry Pi is not designed to be Rad hard, it showed a surprisingly good radiation tolerance in tests (in references, see Violette). It continued to operate through a dose of 150 krad(Si), with only the loss of USB connectivity. An ongoing program by NASA and the Air Force is working on a rad-hard ARM of the Raspberry Pi class.

The Pi Compute Module, update-able to Pi 2 supports a 10 DOF IMU, an RTC, Analog to digital converters, a real time clock, a dedicated camera port, and supports the communication interfaces I2C, SPI, GPIO, Ethernet, USB

Open Source

This is a topic we need to discuss before we go into software. It is not a technical topic, but concerns your right to use (and/or own, modify) software. It's those software licenses you click to agree with, and never read. That's what the intellectual property lawyers are betting on.

Software and software tools are available in proprietary and open source versions. Open source software is free and widely available, and may be incorporated into your system. It is available under license, which generally says that you can use it, but derivative products must be made available under the same license. This presents a problem if it is mixed with purchased, licensed commercial software, or a level of exclusivity is required. Major government agencies such as the Department of Defense and NASA have policies related to the use of Open Source software.

Adapting a commercial or open source operating system to a particular problem domain can be tricky. Usually, the commercial operating systems need to be used "as-is" and the source code is not available. The software can usually be configured between well-defined limits, but there will be no visibility of the internal workings. For the open source situation, there will be a multitude of source code modules and libraries that can be configured and customized, but the process is complex. The user can also write new modules in this case.

Large corporations or government agencies sometimes have problems incorporating open source products into their projects. Open Source did not fit the model of how they have done business traditionally. They are issues and lingering doubts. Many Federal agencies have developed Open Source policies. NASA has created an open source license, the NASA Open Source Agreement (NOSA), to address these issues.

The Open source philosophy is sometimes at odds with the

rigidized procedures evolved to ensure software performance and reliability. Offsetting this is the increased visibility into the internals of the software packages, and control over the entire software package. Besides application code, operating systems such as GNU/linux and bsd can be open source. The programming languages c and Python are open source, along with others.. The popular web server Apache is also open source.

A list of open source tools for Cubesats can be found here:

http://wiki.developspace.net/Open_Source_Engineering_Tools

Cubesat Onboard Software

Onboard software is a special case of embedded software. As such, it is generally more difficult to design, implement, and test. It must be treated carefully, because most of the Cubesat functionality will rely on software, and the mission success will be directly related to software.

Flight Software can be proprietary or Open Source, but almost all Cubesat onboard software is open source.

FSW has several distinguishing characteristics:

- There are no direct user interfaces such as monitor and keyboard. All interactions are through uplink and downlink.

- It interfaces with numerous flight hardware devices such as thrusters, reaction wheels, star trackers, motors, science instruments, temperature sensors, etc.

- It executes on radiation-hardened processors and microcontrollers that are relatively slow and memory-limited.

- It performs real-time processing. It must satisfy numerous timing constraints (timed commands, periodic deadlines,

async event response). Being late = being wrong.

- Besides attitude determination and control, the onboard embedded systems has a variety of housekeeping tasks to attend to.

NASA's Core Flight Executive, and Core Flight Software

The Core Flight Executive, from the Flight Software Branch at NASA/GSFC, is an open source operating system framework. The executive is a set of mission independent reusable software services and an operating environment. Within this architecture, various mission-specific applications can be hosted. The cFE focuses on the commonality of flight software. The Core Flight System (CFS) supplies libraries and applications. Much flight software legacy went into the concept of the cFE. It has gotten traction within the Goddard community, and is in use on many flight projects, simulators, and test beds (FlatSats) at multiple NASA centers, , as well as functioning in on-orbit Cubesat

The cFE presents a layered architecture, starting with the bootstrap process, and including a real time operating system. At this level, a board support package is needed for the particular hardware in use. Many of these have been developed. At the OS abstraction level, a Platform support package is included. The cFE core comes next, with cFE libraries and specific mission libraries. Ap's habituate the 5th, or upper layer. The cFE strives to provide a platform and project independent run time environment.

The boot process involves software to get things going after power-on, and is contained in non-volatile memory. cFE has boot loaders for the RAD750 (from BAE), the Coldfire, and the Leon3 architecture. The real time operating systems can be any of a number of different open source or proprietary products, VxWorks and RTEMS for example. This layer provides interrupt handling, a scheduler, a file system, and interprocess communication.

The Platform Support Package is an abstraction layer that allows the cFE to run a particular RTOS on a particular hardware platform. The cFE Core includes a set of re-usable, mission independent services. It presents a standardized application Program Interface (API) to the programmer. A software bus architecture is provided for messaging between applications.

The Event services at the core level provides an interface to send asynchronous messages, telemetry. The cFE also provides time services.

Aps include a Health and Safety Ap with a watchdog. A housekeeping AP for messages with the ground, data storage and file manager aps, a memory checker, a stored command processor, a scheduler, a check-summer, and a memory manager. Aps can be developed and added to the library with ease.

A recent NASA/GSFC Cubesat project uses a FPGA-based system on a chip architecture with Linux and the cFE. The cFE has been released into the World-Wide Open Source community, and has found many applications outside of NASA.

NASA's software Architecture Review Board reviewed the cFE in 2011. They found it a well thought-out product that definitely met a NASA need. It was also seen to have the potential of becoming a dominant flight software architectural framework. The technology was seen to be mature.

The cFS is the core flight software, a series of aps for generally useful tasks onboard the spacecraft. The cFS is a platform and project independent reusable software framework and set of reusable applications. This framework is used as the basis for the flight software for satellite data systems and instruments, but can be used on other embedded systems in general. More information on the cFS can be found at http://cfs.gsfc.nasa.gov/OSAL

The OS Abstraction Layer (OSAL) project is a small software

library that isolates the embedded software from the real time operating system. The OSAL provides an Application Program Interface (API) to an abstract real time operating system. This provides a way to develop one set of embedded application code that is independent of the operating system being used. It is a form of middleware.

cFS aps

CFS aps are core Flight System (CFS) applications that are plug-in's to the Core Flight Executive (cFE) component. These do specific tasks, such as

the cFS has a layered architecture, allowing one layer to be modified without affecting other layers, This enables technology evolution. It allows platform, operating system, and middleware independence. It includes components such as File Management, Health & Safety, Housekeeping, memory Managemen, and scheduler, among others.

At the moment, there is no encryption/decryption module in Cfs, but one is planned.

How do you send data to and receive data from a Cubesat?

Ok, a couple of hints. Your Cubesat probably has a Raspberry Pi or similar flight computer. It may also have one of several available off-the-shelf transceiver units. You may use a Software Defined Radio module running on the main computer. You may use a shared uplink/downlink service, such as COSMOS. In the latter case, you may want to add your own secure Virtual Private Network.

Internet Protocols

Originally developed for the ARPA-Net, which later became the Internet, these protocols serve us quite well in terrestrial applications. The Internet Protocols have a fatal flaw, however, when it comes to very long distance communication. They were designed for terrestrial applications, of not too long a communication distance. Each packet sent waits for an acknowledgment. If that is not forthcoming in a reasonable period time, the transmitter re-sends the copy. Because the specific path (routing) between the transmitter and receiver is not necessarily known, we can wind-up with a continual repetitive re-sending. It is possible to determine the path with a trace-route, but it will probably be different next time. For long distances, such as from Earth to Mars and beyond, this scheme just fall apart.

CCSDS Protocols

The CCSDS protocols were developed specifically for space use, starting in 1982. They address the use of packet telemetry. CCSDS has an International committee, and is, itself, an ISO subcommittee for Space Data and Information Transfer Systems. It uses Industry standards from the terrestrial Internet as a basis for the space segment.

CCSDS Packets are used on near-Earth and Deep Space communication links. These have more rigorous error detection and correction schemes. The data length is variable, and can be from 7 to 65,542 bytes, which includes the header. Packet sizes are fixed length. The transmission of packets is via data frames, which are also fixed length. The frame also contains control information.

CCSDS defines a number of standards and best practices for satellite communication, including security for the uplink (commands) and the downlink (telemetry). Their “Green Book” focuses on Security for Space Missions. They are working on developing a recommendation for a specific security protocol at the data link layer. It will provide authentication and encryption.

OSI model

The Open Systems Interconnect model defines a standard for communications between units. It is a conceptual model, and defines seven levels or layers, the Application, Presentation, Session, Transport, Network, Data link, and Physical layers. A layer, by definition, serves the layer above it, and is served by the layer below it. The model part of the international Organization for Standardization. Layer 1, the lowest layer, is the physical layer, involving radio or wired communication. Encryption and decryption is usually done at Level 6, the Presentation layer. The definition of the Security Architecture in the OSI model is in their ISO/IEC 7498-2 document.

The Cubesat Space Protocol

For Cubesats, which until very recently were confined to Earth orbit, we would prefer a communications implementation in Open Source. There are implementations of various communications protocols available in Open Source format for the popular Raspberry Pi architecture, and similar units, even the 8-bit AVR. There is a specific Cubesat Space security protocol, based on the same layers as TCP/IP. It does include support for encrypted packets with XTEA in CTR mode.

What are the problems?

When some one else begins to fool around with your Cubesat, there's a lot of harm that can be done. Generally, most builders have assumed they were operating in a benign environment. Who would possibly want to mess around with a Cubesat? Answer: a world wide community of hackers, who also mess around with self-driving cars, traffic lights, the GPS System, Dish Network, and your electric meter. Most of these are not deliberately hostile, but are done “for fun.”

Why do you lock your car or bike? Why do you have a computer password? Because, the world is sometimes hostile. Hack-attacks

against satellites have been going on for years. Now, hackers have smaller, easier targets. In the good old days, a couple of years ago, a high school could build and launch their very own Cubesat, without regard for security. That has changed, and will not change back. What we will talk about here is the nature of attacks against satellites in general, and what can be done to prevent these.

At one point, we left embedded computer devices onto our Internet. This has made it easier to mount large scale attacks. Your “smart” electric meter can be hacked to shut down your electricity. Security systems cameras can be hacked to provide potential burglars instant access. Have a smart lock? We can hack that. The author had an incident where with two cars parked in the driveway, different manufacturers, both use key fobs, both were unlocked and rummaged through. The police are not on top of this yet, but the internet has information on the devices you need to buy, and from whom. Felling better now? Putting the car in the garage? Did I mention hacking automatic garage doors?

Nation States routinely hack their opponents space assets to see where the vulnerability's are. A deep-pocket hacker, not necessarily more than a millionaire, could launch his own hacker-sat to orbit. Having trouble getting to sleep at night yet?

Security, like safety, can rarely be added on. It needs to be considered and implemented from the very beginning. In addition, we know ways to protect data, usually encryption. But, we also have to worry about transmission security. A major issue in encryption and decryption is the security of keys. These are the code sequences used to encrypt and decrypt data. They have to be kept in tight security. The longer the key, the better the security, but also the greater overhead. Keys are generally generated as a random number. They should be changed frequently, which is another overhead item.

Spacecraft Cybersecurity Policy for non-DoD payloads is controlled by the sponsors, NASA, NOAA (for remote sensing missions), and the FAA for private missions. Each of these entity's has their own policys. Commercial payloads fall under the

National Commercial and Space Programs Act, with licensing authority given to NOAA for imager missions. NASA has no current requirement to encrypt. Private, non-imaging concerns operate under the lack of co-ordinated public policy in this matter.

Security

Has your Cubesat project been hacked? Are you sure? Security applies to the control center, the users, and the embedded controllers onboard the Cubesat.

There are a lot of bored and/or malicious computer experts out there, ranging from State-sponsored to idle teenagers at an Internet Cafe. They all want to play with your Cubesat. Just like anything with a computer, we can have hacking, virus and malware insertion, theft, spoofing, disruption, denial of service, and your Cubesat or control center can be used as a launch point for denial of service attacks against other systems. This has happened to traffic light controllers, ATM's, elevators, trains, planes, and automobiles, and implanted medical devices.

Don't ask if your systems is going to be attacked. Assume it will be, and prevent it by Design.

All systems have aspects of security. Some of these issues are addressed by existing protocols and standards for access and communications security. Security may also imply system stability and availability. Standard security measures such as security reviews and audits, threat analyses, target and threat assessments, countermeasures deployment, and extensive testing apply.

Addressing security involves:

- Use of industry standards and best practices.
- security audits.
- threat analysis.
- target assessment.

- Countermeasures.
- testing, testing, testing

A security assessment of a system involves threat analysis, target assessment, risk assessment, countermeasures assessment, and testing. This is above and beyond basic system functionality.

The completed functional system may need additional security features, such as intrusion detection, data encryption, and perhaps a self-destruct capability. Is that self-destruct capability secure, so not just anyone can activate it? All of these additional features use time, space, and other resources that are usually scarce in embedded systems.

Virus and malware attacks on desktops and servers are common, and an entire industry related to detection, prevention, and correction has been spawned. These issues are not as well addressed in the embedded world. Attacks on new technology such as cell phones, tablets, and GPS systems are emerging. Not all of the threats come from individuals. Some are large government-funded efforts or commercial entities seeking proprietary information or market position. Security breaches can be inspired by ideology, money, or fame considerations. The *CERT* (Computer Emergency Response Team) organization at Carnegie Mellon University, and the *SANS* Institute (SysAdmin, Audit, Networking, and Security) track security incidents.

Techniques such as hard check-sums and embedded serial numbers are one approach to device protection. Access to the system needs to be controlled. If unused ports exist, the corresponding device drivers should be disabled, or not included. Mechanisms built into the cpu hardware can provide protection of system resources such as memory.

Security has to be designed in from the very beginning; it can't just be added on. Memorize this. There will be a quiz. Control center

security begins with the entry points, where cables enter and exit the room. It continues with a good lock on the door, and the allowed access of certified personnel. Running good intrusion detection on the support computers is essential. But the hacker does not need to penetrate your physical security. He or she can talk directly to your satellite form his or her own ground station.

Even the most innocuous embedded platform can be used as a springboard to penetrate other systems. It is essential to consider security of all embedded systems, be aware of industry best practices and lessons learned, and seek professional help in this specialized area.

The Satellite Control Center must quickly act quickly to identify and react to attacks. These might be phishing expeditions to steal data, malicious attacks to gain control of space assets, or denial of service attacks. A good penetration-testing of the facility is called for. There should be a permanent Security Officer, with the responsibility for data and operations.

Issues:

- access control
- system stability
- intrusion detection
- data encryption
- time, space, other resources scarce

Denial of Service

In a Denial of Service scenario, the bad guys take over a system, and make it unavailable to the owner. This has been done numerous times, including to the satellite assests of DishNetwork, the TV provider. Parts of Wikipedia's website were shut down by a DoS attack in 2019.

A distributed denial of service attack comes from multiple locations, and is harder to counter. DDS attacks are used to disrupt

transactions or deny communications and database access. Another effect is the starvation of resources approach. The longest attack recorded lasted 38 days. There are now standard attack tools, available for download on the Internet. In the United States, at least, a DoS attack is a Federal crime, investigated by the Department of Justice.

One approach to countering DoS attacks is hardware-based, and is less appealing to small systems than a software approach.

Very soon, any satellite that has a propulsion system will be required to have encryption. That is because it could be hijacked and used to interfere with or damage other spacecraft. It is unclear if this includes solar-sail and Ion-engine equipped units.

A DoS attach on a Cubesat would disrupt its operation, and make it unusable to its owner. Valuable data would be lost. A DoS attack, in the form of Ransom-ware that would only accept untraceable BitCoin, could be mounted against Cubesats.

Soon, the launch authority will require encryption on all space mission uplinks. (the command side). For Cubesats, this will apply to all units with propulsion systems. One area this addresses is the weaponization of the Cubesats, using them to damage other spacecraft. Arguments are that Cubesats have, if any, solar sail or ion thruster propulsion, and these are fairly weak systems. This policy is referred to as the “no encryption, no fly” rule. The bi-propellant systems are of the most concern, followed by cold-gas, ion drive, and solar sail. We do not know at the moment whether a Cubesat has ever been hacked. Regulatory agency are hoping the Cubesat industry chooses to regulate itself.

Encryption of the uplink will prevent most attacks against Cubesats, in attempts to reprogram them. Encryption of the downlink will probably lag behind, as this only effects theft of data.

How do you know your Cubesat has been hacked? How do you know your computer, tablet, or phone has been hacked? It slows down, and stops listening to you. It becomes unresponsive. There

may be an explicit message that you have been hacked, and defines the ransom to be provided, and how

There is a lot of security built into our devices, based on past problems. We haven't done this for our Cubesats yet. There's a smart teenager in some third-world coffee shop that enjoys flying your Cubesat. Just as in Europe, where a bored, tech-savvy teenager hacked into and took over control of his City's tram system for fun. A great big model train layout.

Software Defined Radio

Software Defined Radio involves using an embedded computer to modulate and demodulate a signal. The RF receiver will produce an intermediate frequency, that is then fed through an analog to digital converter. It then goes to the flight computer to be decoded. The flight computer on a Cubesat is usually adequate for this. Encryption/description adds another layer of complexity, and further burden on the processor.

Uplink

To send the malicious code to the Cubesat, you need a satellite uplink station. You are imaging a big 30 meter disk like NASA has. Well, actually, like all things open source, the design, hardware, and software details for a much smaller system are available on the Internet. They use a Yagi antenna, like old fashion tv antennas. The electronics is an ARM-based Software Defined Radio. You're ready to go. Oh, forgot licensing. In the U.S., the Federal Communications Commission requires you to have a license before broadcasting. You must comply with that. There are specific frequency bands for Cubesat use. If the signal is encoded, the next step is for the computer to provide error detection and correction, if applicable. All of this takes time and compute cycles.

What are the solutions?

Our first approach to making sure our Cubesat is listening to us and not anyone else is to encrypt the uplink. At the moment we will not worry about encrypting the downlink. That's data coming down, and the project is open source, right? We just want to be in charge of our own Cubesat.

So, using some industry standard encryption means that the transmission and decode it on the Cubesat will have more overhead. But, uplink tends to be short messages containing commands for the Cubesat, so the impact of encryption will be light.

Encryption can be done at the Ground Station, before uplink, using an additional module for the Software Defined Radio.

Up until recently, no one really thought hacking into a Cubesat would be attempted. It has happened, and provides a new challenge for extremely good technical people with time on their hands, or for state actors. NASA, a high profile organization, has had to deal with this situation for years. Beyond attempting to send commands to a target, a hacker can jam the uplink, denying receipt of legitimate commands, with sufficient ground station power. This would certainly be noticeable.

What do you need to hack a satellite? Well, an uplink antenna, and a transmitter. Then you need to know where the satellite is. NORAD will help you with that. NORAD tracks each and every piece in orbit, from the ISS, to smaller chunks of space debris.

You do have to know the Satellite's ID number, assigned at launch.

Cubesats can be tracked by radar. NORAD, the North American Aerospace Command, based in Colorado, tracks all detectable orbital entities, from large satellites to space junk, zombie-sats, and the larger pieces of debris, as well as near-Earth asteroids.

NORAD puts all this up on a website for your convenience, in a standard format called the “two-line element” (TLE). This contains the Keplerian orbital elements, the set of data describing the orbit

of anything around the Earth, for a given point in time (epoch). It is a legacy format form the 1960's, that still works. It includes two data items of 80 ASCII charters each (an IBM punch card format).

Here is the format and contents of Line 1:.

Field	Columns	Content
1	1	Line number
2	3-7	Satellite number
3	8	Classification (U = unclassified)
4	10-11	Internat. Designator, last two digits of launch year
5	12-14	Launch number of the year
6	15-17	Place of launch
7	19-20	Epoch, last two digits of year
8	21-32	Epoch, day of year, and fractional portion of day
9	34-43	First Time Derivative of the Mean Motion divided by two
10	45-52	Second Time Derivative of Mean Motion divided by six (decimal point assumed)
11	54-61	BSTAR drag term, (decimal point assumed)
12	63	number 0
13	65-68	element set number, incremented for new TLE
14	69	Checksum, modulo 19

Here is the format of line 2:

Field	Columns	Content
1	1	Line number
2	3-7	Satellite number
3	9-16	Inclination (degrees)
4	18-25	Right ascension of the ascending node (degrees)
5	27-33	Eccentricity (decimal point assumed)
6	35-42	Argument of perigee (degrees)
7	44-51	Mean Anomaly (degrees)

8	53-63	Mean Motion (revolutions per day)
9	64-68	Revolution number at epoch (revolutions)
10	69	Checksum (modulo 10)

Ok, drop these data into your orbital simulator, and it will tell you when the satellite is visible from yourcubesat space p location, and where to point the antenna.

You can also get an account with spacetrack.org, or use this service:

http://www.celestrak.com/NORAD/elements/

There are many Open Source data communications security solutions out there. OpenSSH, and Open Secure Shell, provides a secure channel over an unsecured network. It runs under Open BSD, and Linux. GnuPG, GNU Privacy Guard, is based on Semantec's PGP software. It supports Public Key encryption.

Security at the Control Center

The Control Center for the Cubesat mission has issues of security, both operational and data. First, it needs to be a controlled access facility. Only authorized personnel should be in the Control Center facility But don't forget virtual users. Can the control center computers be hacked into? What ports and backdoors exist in the implementation? Best practices from the Computer/Network Security Industry should be applied. Lose your laptop, lose your Cubesat.

Many Cubesat Control Centers are on college campuses, where physical security is minimal The Control Center must quickly act quickly to identify and react to counterattacks. These might be phishing expeditions to steal data, malicious attacks to gain control of space assets, or denial of service attacks. A good penetration-testing of the facility should be conducted. There should be a permanent Security Officer, with the responsibility for data and

operations, and getting in touch with the relevant agency's when trouble is detected.

Afterword

So, why are you still reading this book, and not hardening your Cubesat against the bad guys? I'll can you how to do it, but I won't do it for you. You can put it off for a while, and become an international news phenom, smeared all over Facebook. This stuff is real, address it.

Glossary of Terms

1U – one unit for a Cubesat, 10 x 10 x 10 cm.
AES – Advanced Encryption Standard., NIST.
AFSCN – (U. S.) Air Force Satellite Control Network.
AI – artificial intelligence.
Android – an operating system based on Gnu-Linux, popular for smart phones and tablet computers.
AntiVirus – a computer program to prevent, detect, and remove viruses.
API – Application Program Interface.
Arduino – a small, inexpensive microcontroller architecture.
Arinc – Aeronautical Radio, Inc. commercial company supporting transportation, and providing standards for avionics.
ARM – Acorn RISC machine; a 32-bit architecture with wide application in embedded systems.
ARP – Address resolution protocol.
ARPA – (U. S.) Advanced Research Projects Agency, became DARPA.
Anonymous – (Group) lose confederation of hacktivists.
async – non synchronized.
Baud – symbol rate; may or may not be the same as bit rate.
BIST – built-in self test.
Bitcoin – digital cryptocurrency.
Bootloader – initial program run after power-on or reset. Gets the computer up & going.
Bot – a software robot.
Botnet – network of bots.
BP - bundle protocol, for dealing with errors and disconnects.
BSD – Berkeley System Division, port of Bell labs Unix.
BSL – Bundle Service Layering
CalPoly – California Polytechnic State University,. San Luis Obispo, CA.
CCITT – International Telegraph and Telephone Consultative Committee. Acronym is from the French version.
CCSDS – Consultive Committee on Space Data Systems.

CDR – critical design review
C&DH – Command and Data Handling
CDFP - CCSDS File Delivery Protocol
CEH – Certified Ethical Hacker, a qualification.
CERT – Computer Emergency Response Team (SEI)
cFE – Core Flight Executive – NASA GSFC reusable flight software.
CFS – Core Flight System – NASA GSFC reusable flight software.
Cipher – algorithm for encrypting or decrypting.
Code – conversion of symbols fro easier or shorter transmission, or to hide the message
Cookie - small piece of data from a website, stored on a user's computer.
COP – computer operating properly.
Cosmos – open source control center software from Ball Brothers Aerospace.
Cots – commercial, off the shelf.
CRC – cyclic redundancy code – error detection and correction mechanism.
CSP – Cubesat Space Protocol.
CSRF – Cross-site Request Forgery.
CTR – Content threat removal.
Cubesat – small inexpensive satellite for colleges, high schools, and individuals.
Cybersecurity – Computer or IT security.
Cryptovirology- how to use cryptography to design powerful malicious software.
Cypher – (also, Cipher) – encoded.
ARPA – (U. S.) Defense Advanced Research Projects Agency.
DOS – denial of service
DDOS – distributed denial of service.
Decryption - un-encoding encrypted data.
DOT – dictionary of terms.
DTN – Delay or Disruption Tolerant Network.
Easter Egg – data hidden within a video game, movie, or other electronic data.

Encryption – encoding a message so only authorized recipients can read it.
EOL – end of life.
ESA – European Space Organization.
ESRO – European Space Research Organization
ESTO – NASA/GSFC – Earth Science Technology Office.
ESA – European Space Organization.
ESRO – European Space Research Organization
ESTO – NASA/GSFC – Earth Science Technology Office.
FAA – (U.S.) Federal Aviation Administration.
FCC – (U.S.) Federal Communications Commission.
FCS – frame check sequence, for error detection.
FDC – fault detection and correction.
Frame – fixed length data block.
GCHQ – (UK) -Government Communications Headquarters.
GHz – giga (10^9) hertz
GNC – guidance, navigation, and control.
Gnu – recursive acronym, gnu is not unix.
GNUPG – GNU Privacy Guard, open source replacement for PGP.
GPS – Global Positioning system – Navigation satellites.
Hacker – skilled computer user, with advanced skills.
Hacktivist – hack activism, using technology to promote social or political ideas.
HMAC – hash-based message authentication code.
IARU – International Amateur Radio Union.
IEFT – Internet Engineering Task Force.
IF – intermediate frequency.
IFG – inter-frame gap.
IKE – Internet key exchange.
ION – (NASA) Interplanetary Overlay Network.
IOT – Internet of things.
IP – intellectual property; Internet protocol.
IPSec – Internet Protocol Security
IP-in-Space – Internet Protocol in Space.
IRTF – Internet Research Task Force.
ISO – International Standards Organization.
ISS – International Space Station.

IT – Information Technology
I&T – integration & test.
ITU – International Telecommunications Union
IV&V – Independent validation and verification.
Jamming – interfering with a data transmission.
JHU – Johns Hopkins University.
JPL – Jet Propulsion Laboratory
Kernel – main portion of the operating system. Interface between the applications and the hardware.
LEO – low Earth orbit.
LRR – launch readiness review.
Malware – malicious software.
Memory scrubbing – detecting and correcting bit errors.
Microkernel – operating system which is not monolithic, functions execute in user space.
MMU – memory management unit; manned maneuvering unit.
Multicore – multiple processing cores on one substrate or chip; need not be identical.
Mutex – a software mechanism to provide mutual exclusion between tasks.
NanoSat – small satellite with a mass between 1 and 10 kg.
NASA - National Aeronautics and Space Administration.
NIDS – Network Intrusion Detection System.
NIST – (U.S.) National Institute of Standards and Technology.
NORAD – (U. S.) North American Aerospace Defense Command.
NSA – (US) National Security Agency, also, no-such agency.
OBC – on board computer
OBD – On-Board diagnostics.
ODM – (U.S.) Orbital Debris Mitigation Standard Practices.
OBP – On Board Processor
Open source – methodology for hardware or software development with free distribution and access.
OpenSSH – Open Secure Shell, crypto network protocol for security over an unsecured network.
Operating system – software that controls the allocation of resources in a computer.
OSAL – operating system abstraction layer.

OSCAR - Orbiting Satellite Carrying Amateur Radio.
OSI – Open Source Initiative, Open Source Institute, Open Systems Interconnect, circa defined 1978.
OSINT – Open Source Intelligence – information collected from public sources.
Packet – a small, formatted unit of data.
PDR – preliminary design review.
PDU – protocol data unit.
PGP – pretty good privacy, an encryption/decryption protocol
Phishing – an attempt to obtain sensitive information fraudulently.
PiSat – a Cubesat architecture developed at NASA-GSFC, based on the Raspberry Pi architecture.
Plain text – you're looking at it.
PocketQube – smaller than a Cubesat; 5 cm cubed, a mass of no more than 180 grams, and uses COTS components.
P-POD – Cubesat launch dispenser, Poly-Picosatellite Orbital Deployer.
PRNG - pseudorandom number generator.
QKD = Quantum key distribution.
Quantum cryptography – theoretically unbreakable cryptography, using a shared key.
Quantum hacker – breaking quantum cryptography.
Ransomeware – malware that will expose a user's sensitive data unless a ransom is paid.
RBF – remove before flight.
Real-time – system that responds to events in a predictable, bounded time.
Root kit – a usually malicious collection of software to allow one to take over a computer without authorization.
RTC – real time clock.
RTOS – real time operating system.
Sandbox – an isolated and controlled environment to run untested or potentially malicious code.
SatIPSec – securing satellite transmissions.
SATNOGS - Satellite Networked Open Ground Station.
SCPS – Space Communications Protocol Specification.
SDR – software defined radio

SDRAM – synchronous dynamic ram.
SDU – service data unit.
SDVF – Software Development and Validation Facility.
SEI – Software Engineering Institute, CarnegieMellon University.
SIGINT – signals intelligence.
Six-pack – a six U Cubesat, 10 x 20 x 30 cm.
SMS – short message service.
Sniffer – intercepts and log traffic over a network.
Snoop – monitor packets in a network, or data in a cache.
Snort - Network Intrusion Detection & Prevention System
Spacewire – high speed (160 Mbps) link.
Space-X – commercial space company.
TCP/IP – Transmission Control Protocol/Internet protocol.
Spoofing – pretending to be a different sender.
Spyware – software to gain information.
SSH – secure shell, a cryptographic network protocol.
SSI – solar system internet.
SSL – secure socket layer
TCP/IP – transmission control protocol/internet protocol; layered set of protocols for networks.
TDRSS – Tracking and Data Relay satellite system.
TFS – Traffic flow security
T&I – test and integration.
TLE - Three line element defining a satellites orbital parameters.
Triplicate – using three copies (of hardware, software, messaging, power supplies, etc.). for redundancy and error control.
TRL – technology readiness level.
Trojan Horse – malware hidden within innocuous code.
UHF – ultra high frequency (300 MHz – 3 GHz)
Uplink – data from the ground to the satellite.
URL – Universal Resource Locator.
USAF – United States Air Force.
VHF – very high frequency (30-300 MHz)
Virus – malignant computer program.
VPN – virtual private network.
Weaponize – using some common thing as a weapon.
Wireshark – network protocol analyzer.

X25– a packet-switched communication protocol

XML – extensible Mark-up language, defines rules for encoding documents to be human and machine readable.

XTEA – extended tiny encryption algorithm.

Zombie-sat – a dead satellite, in orbit.

ZOE - Zone of Exclusion, volume in which the presence of an object or personnel, or activities are prohibited

Bibliography

Antunes, Sandy *DIY Comms and Control for Amateur Space: Talking and Listening to Your Satellite,* ISBN 978-1449310660.

Barron, Andrew *Amsats and Hamsats: Amateur Radio and other Small Satellites,* 2018, ASIN-B07CG8DYR2.

Burleigh, Scott C. et al. From Connectivity to Advanced Internet Services: A Comprehensive Review of Small Satellites Communications and Networks, avail: https://www.hindawi.com/journals/wcmc/2019/6243505/

Dittrich, David, Reiher,Peter et al. *Internet Denial of Service: Attack and Defense Mechanisms,* ISBN-978-0131475731.

Duquerroy, Laurence, et al *SatiPSec : An Optimized Solution for Securing Multicast and Unicast Satellite Transmissions, 2004, avail:* https://www.researchgate.net/publication/29608892_SatiPSec_An_Optimized_Solution_for_Securing_Multicast_and_Unicast_Satellite_Transmissions

Engebretson, Patrick *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy,* 2013, ISBN-9780124116443.

Erickson, Jon *Hacking: The Art of Exploitation,* 2008, ISBN-978-1593271442.

Hudaib, Adam *Satellite Network Threats Hacking & Security Analysis: Satellite Network Hacking Security Analysis ,Threats and Attacks , Architecture Operation design and technologies,* ISBN-978-1535252546.

Mitnick, Kevin D., Simon, William L. *The Art of Intrusion: The*

Real Stories Behind the Exploits of Hackers, Intruders and Deceivers, 2005, ISBN-978-0471782667.

NASA, *An Assessment of Smallsat Technology to Future Exploration Missions*, 2019, NASA CR-202345 ASIN-B07WQJ3R54.

Neilson, Seth *Practical Cryptography in Python: Learning Correct Cryptography by Example,* 2019, ISBN-978-1484248997.

Özçelik, İlker *Distributed Denial of Service Attacks: Real-world Detection and Mitigation,* ISBN-978-0367491543.

Parr, Christof, et al *Understanding Cryptography: A Textbook for Students and Practitioner,* 2010, ISBN- 978-3642041006.

Pelton, Joseph N. *Handbook of Small Satellites: Technology, Design, Manufacture, Applications, Economics and Regulation,* 1st ed., 2020, ISBN-978-3030363079.

Schneier, Bruce *Applied Cryptography: Protocols, Algorithms and Source Code in C,* ISBN- 978-1119096726.

Sims, Eleni M. ; Braun, Barbara M., *Navigating the Policy Compliance Raodmap for Sam;; Satellites.* Aerospace Corp., Center for space Policy and Strategy, 2017. avail: https://aerospace.org/sites/default/files/2018-05/SmallSatRegulations_0.pdf

Stallings, William *Network Security Essentials: Applications and Standards (6th Edition),* ISBN-978-0134527338.

Stallings, William *Computer Security: Principles and Practice,* ISBN-978-0133773927.

Talleur, Thomas J. (18 January 1999). Russian Domain Attacks Against NASA Network Systems. Not publicly published.

Classified as "For Official Use Only—No Foreign Dissemination": Inspector General's office, NASA, 26.

Tannenbaum, Andrew S. *Computer Networks*, 4th ed, ISBN 0-13-066102-3.

Wagner, Alex *Hacking: Denial of Service Attacks,* 2019, ASIN-B082VH35T8.

Walker, Matt *CEH Certified Ethical Hacker Bundle, Fourth Edition,* ISBN-978-1260455267.

Resources

https://astronomy.com/news/2020/02/hackers-could-shut-down-satellites--or-turn-them-into-weapons

https://theconversation.com/hackers-could-shut-down-satellites-or-turn-them-into-weapons-130932 .

Invaders from space — hacks against satellites threaten our critical infrastructure
https://www.sfchronicle.com/opinion/article/Invaders-from-space-hacks-against-satellites-13175362.php

Anti-Phishing Working Group, APWG.org

https://www.researchgate.net/publication/228712402_On-board_security_services_in_small_satellites

https://www.satellitetoday.com/telecom/2005/06/01/satellite-and-secure-communications-a-strategic-combination/

www.CCSDS.org

wikipedia, various.

If you enjoyed this book, you might also be interested in some of these.

Stakem, Patrick H. *16-bit Microprocessors, History and Architecture*, 2013 PRRB Publishing, ISBN-1520210922.

Stakem, Patrick H. *4- and 8-bit Microprocessors, Architecture and History*, 2013, PRRB Publishing, ISBN-152021572X,

Stakem, Patrick H. *Apollo's Computers,* 2014, PRRB Publishing, ISBN-1520215800.

Stakem, Patrick H. *The Architecture and Applications of the ARM Microprocessors,* 2013, PRRB Publishing, ISBN-1520215843.

Stakem, Patrick H. *Earth Rovers: for Exploration and Environmental Monitoring,* 2014, PRRB Publishing, ISBN-152021586X.

Stakem, Patrick H. *Embedded Computer Systems, Volume 1, Introduction and Architecture*, 2013, PRRB Publishing, ISBN-1520215959.

Stakem, Patrick H. *The History of Spacecraft Computers from the V-2 to the Space Station*, 2013, PRRB Publishing, ISBN-1520216181.

Stakem, Patrick H. *Floating Point Computation*, 2013, PRRB Publishing, ISBN-152021619X.

Stakem, Patrick H. *Architecture of Massively Parallel Microprocessor Systems*, 2011, PRRB Publishing, ISBN-1520250061.

Stakem, Patrick H. *Multicore Computer Architecture,* 2014, PRRB

Publishing, ISBN-1520241372.

Stakem, Patrick H. *Personal Robots*, 2014, PRRB Publishing, ISBN-1520216254.

Stakem, Patrick H. *RISC Microprocessors, History and Overview,* 2013, PRRB Publishing, ISBN-1520216289.

Stakem, Patrick H. *Robots and Telerobots in Space Application*s, 2011, PRRB Publishing, ISBN-1520210361.

Stakem, Patrick H. *The Saturn Rocket and the Pegasus Missions, 1965,* 2013, PRRB Publishing, ISBN-1520209916.

Stakem, Patrick H. *Visiting the NASA Centers, and Locations of Historic Rockets & Spacecraft,* 2017, PRRB Publishing, ISBN-1549651205.

Stakem, Patrick H. *Microprocessors in Space*, 2011, PRRB Publishing, ISBN-1520216343.

Stakem, Patrick H. Computer *Virtualization and the Cloud*, 2013, PRRB Publishing, ISBN-152021636X.

Stakem, Patrick H. *What's the Worst That Could Happen? Bad Assumptions, Ignorance, Failures and Screw-ups in Engineering Projects, 2014,* PRRB Publishing, ISBN-1520207166.

Stakem, Patrick H. *Computer Architecture & Programming of the Intel x86 Family, 2013,* PRRB Publishing, ISBN-1520263724.

Stakem, Patrick H. *The Hardware and Software Architecture of the Transputer*, 2011,PRRB Publishing, ISBN-152020681X.

Stakem, Patrick H. *Mainframes, Computing on Big Iron*, 2015, PRRB Publishing, ISBN- 1520216459.

Stakem, Patrick H. *Spacecraft Control Centers*, 2015, PRRB Publishing, ISBN-1520200617.

Stakem, Patrick H. *Embedded in Space,* 2015, PRRB Publishing, ISBN-1520215916.

Stakem, Patrick H. *A Practitioner's Guide to RISC Microprocessor Architecture*, Wiley-Interscience, 1996, ISBN-0471130184.

Stakem, Patrick H. *Cubesat Engineering*, PRRB Publishing, 2017, ISBN-1520754019.

Stakem, Patrick H. *Cubesat Operations*, PRRB Publishing, 2017, ISBN-152076717X.

Stakem, Patrick H. *Interplanetary Cubesats*, PRRB Publishing, 2017, ISBN-1520766173 .

Stakem, Patrick H. Cubesat Constellations, Clusters, and Swarms, Stakem, PRRB Publishing, 2017, ISBN-1520767544.

Stakem, Patrick H. *Graphics Processing Units, an overview*, 2017, PRRB Publishing, ISBN-1520879695.

Stakem, Patrick H. *Intel Embedded and the Arduino-101, 2017,* PRRB Publishing, ISBN-1520879296.

Stakem, Patrick H. *Orbital Debris, the problem and the mitigation*, 2018, PRRB Publishing, ISBN-*1980466483.*

Stakem, Patrick H. *Manufacturing in Space*, 2018, PRRB Publishing, ISBN-1977076041.

Stakem, Patrick H. *NASA's Ships and Planes*, 2018, PRRB Publishing, ISBN-1977076823.

Stakem, Patrick H. *Space Tourism*, 2018, PRRB Publishing, ISBN-

1977073506.

Stakem, Patrick H. *STEM – Data Storage and Communications*, 2018, PRRB Publishing, ISBN-1977073115.

Stakem, Patrick H. *In-Space Robotic Repair and Servicing*, 2018, PRRB Publishing, ISBN-1980478236.

Stakem, Patrick H. *Introducing Weather in the pre-K to 12 Curricula, A Resource Guide for Educators*, 2017, PRRB Publishing, ISBN-1980638241.

Stakem, Patrick H. *Introducing Astronomy in the pre-K to 12 Curricula, A Resource Guide for Educators*, 2017, PRRB Publishing, ISBN-198104065X.
Also available in a Brazilian Portuguese edition, ISBN-1983106127.

Stakem, Patrick H. *Deep Space Gateways, the Moon and Beyond*, 2017, PRRB Publishing, ISBN-1973465701.

Stakem, Patrick H. *Exploration of the Gas Giants, Space Missions to Jupiter, Saturn, Uranus, and Neptune*, PRRB Publishing, 2018, ISBN-9781717814500.

Stakem, Patrick H. *Crewed Spacecraft*, 2017, PRRB Publishing, ISBN-1549992406.

Stakem, Patrick H. *Rocketplanes to Space*, 2017, PRRB Publishing, ISBN-1549992589.

Stakem, Patrick H. *Crewed Space Stations,* 2017, PRRB Publishing, ISBN-1549992228.

Stakem, Patrick H. *Enviro-bots for STEM: Using Robotics in the pre-K to 12 Curricula, A Resource Guide for Educators,* 2017, PRRB Publishing, ISBN-1549656619.

Stakem, Patrick H. *STEM-Sat, Using Cubesats in the pre-K to 12 Curricula, A Resource Guide for Educators*, 2017, ISBN-1549656376.

Stakem, Patrick H. *Embedded GPU's*, 2018, PRRB Publishing, ISBN- 1980476497.

Stakem, Patrick H. *Mobile Cloud Robotics*, 2018, PRRB Publishing, ISBN- 1980488088.

Stakem, Patrick H. *Extreme Environment Embedded Systems,* 2017, PRRB Publishing, ISBN-1520215967.

Stakem, Patrick H. *What's the Worst, Volume-2*, 2018, ISBN-1981005579.

Stakem, Patrick H., *Spaceports*, 2018, ISBN-1981022287.

Stakem, Patrick H., *Space Launch Vehicles*, 2018, ISBN-1983071773.

Stakem, Patrick H. *Mars*, 2018, ISBN-1983116902.

Stakem, Patrick H. *X-86, 40th Anniversary ed*, 2018, ISBN-1983189405.

Stakem, Patrick H. *Lunar Orbital Platform-Gateway*, 2018, PRRB Publishing, ISBN-1980498628.

Stakem, Patrick H. *Space Weather*, 2018, ISBN-1723904023.

Stakem, Patrick H. *STEM-Engineering Process*, 2017, ISBN-1983196517.

Stakem, Patrick H. *Space Telescopes,* 2018, PRRB Publishing, ISBN-1728728568.

Stakem, Patrick H. *Exoplanets*, 2018, PRRB Publishing, ISBN-9781731385055.

Stakem, Patrick H. *Planetary Defense*, 2018, PRRB Publishing, ISBN-9781731001207.

Patrick H. Stakem *Exploration of the Asteroid Belt*, 2018, PRRB Publishing, ISBN-1731049846.

Patrick H. Stakem *Terraforming*, 2018, PRRB Publishing, ISBN-1790308100.

Patrick H. Stakem, *Martian Railroad,* 2019, PRRB Publishing, ISBN-1794488243.

Patrick H. Stakem, *Exoplanets,* 2019, PRRB Publishing, ISBN-1731385056.

Patrick H. Stakem, *Exploiting the Moon,* 2019, PRRB Publishing, ISBN-1091057850.

Patrick H. Stakem, *RISC-V, an Open Source Solution for Space Flight Computers,* 2019, PRRB Publishing, ISBN-1796434388.

Patrick H. Stakem, *Arm in Space*, 2019, PRRB Publishing, ISBN-9781099789137.

Patrick H. Stakem, *Extraterrestrial Life*, 2019, PRRB Publishing, ISBN-978-1072072188.

Stakem, Patrick H. Submarine Launched Ballistic Missiles, 2019, ISBN-978-1088954904.

Patrick H. Stakem, *Space Command*, Military in Space, 2019, PRRB Publishing, ISBN-978-1693005398.

2020 Releases

CubeRovers, a synnergy of Technologys

Exploration of Lunar & Martian Lava Tubes by Cube-X

Robotic Exploration of the Icy Moons of the Gas Giants.

Hacking Cubesats

History and Future of Cubesats.